Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) An in-circuit emulation system, comprising:

a microcontroller, wherein said microcontroller sends I/O read data to a virtual microcontroller, and wherein said I/O read data is followed by a conditional jump instruction;

a virtual microcontroller coupled to ~~and executing instructions in lock-step with~~ the microcontroller ~~by executing the same instructions using the same clocking signals, and wherein the microcontroller sends I/O read data to the virtual microcontroller~~, wherein said virtual microcontroller ~~the virtual microcontroller having~~ has means for detecting ~~an~~ said I/O read data ~~instruction followed by a conditional jump instruction~~, and further has means for computing a speculative conditional jump address before a condition for said conditional jump instruction is satisfied ~~prior to~~ after receipt of ~~said conditional jump instruction~~ said I/O read data ~~from the microcontroller to remain in lockstep execution with said microcontroller~~; and

the virtual microcontroller further having means for determining after receipt of the I/O read data from the microcontroller whether to proceed with

instruction execution at a next consecutive address or at the speculative conditional jump address, wherein said virtual microcontroller executes instructions at said next consecutive address or at said speculative conditional jump address based on said means for determining, using the same clocking signal, such that said microcontroller and said virtual microcontroller remain in lockstep.

2. (original) The apparatus according to claim 1, wherein the conditional jump address is computed while the I/O read data are sent from the microcontroller to the virtual microcontroller.

3. (original) The apparatus according to claim 1, wherein the microcontroller sets a zero flag if an I/O read test condition is met.

4. (original) The apparatus according to claim 3, wherein the jump condition is met if the zero flag is set.

5. (original) The apparatus according to claim 1, wherein the virtual microcontroller is implemented in a Field Programmable Gate Array.

6. (currently amended) ~~In an in-circuit emulation system having a microcontroller coupled to and operating in lock-step with a virtual~~

~~microcontroller, a~~ A method of handling conditional jumps in ~~the~~ a virtual microcontroller operating in lock-step with a microcontroller, comprising:

detecting an I/O read data ~~instruction followed immediately by a conditional jump instruction~~ sent by said microcontroller, wherein said I/O read data is followed immediately by a conditional jump instruction;

after receipt of said I/O read data, computing a speculative conditional jump address before a condition for said conditional jump instruction is satisfied, ~~prior to receipt of said conditional jump instruction from the microcontroller to remain in lockstep execution with said microcontroller~~; ~~and~~

~~determining~~ after receipt of the I/O read data from the microcontroller, determining whether a conditional jump condition is met; and

executing instruction based on said determination, such that said virtual microcontroller remains in lock-step execution with said microcontroller.

7. (currently amended) The ~~in-circuit emulation system~~ method according to claim 6, ~~further comprising~~ wherein said executing comprises execution of [[a]] next consecutive instruction in the event said conditional jump condition is not met.

8. (currently amended) The ~~in-circuit emulation system~~ method according to claim 6, ~~further comprising~~ wherein said executing comprises execution of an

instruction at the speculative conditional jump address in the event the conditional jump condition is met.

9. (currently amended) The ~~in-circuit emulation system~~ method according to claim 6, wherein the conditional jump address is computed while the I/O read data are sent from the microcontroller to the virtual microcontroller.

10. (currently amended) The ~~in-circuit emulation system~~ method according to claim 6, wherein the microcontroller sets a zero flag if an I/O read test condition is met.

11. (currently amended) The ~~in-circuit emulation system~~ method according to claim 10, wherein the jump condition is met if the zero flag is set.

12. (currently amended) The ~~in-circuit emulation system~~ method according to claim 6, wherein the virtual microcontroller is implemented in a Field Programmable Gate Array.

13. (currently amended) The ~~in-circuit emulation system~~ method according to claim 6, wherein instructions are stored in an electronic storage medium for execution as program steps on a programmed processor forming a part of the virtual microcontroller.

14. (currently amended) ~~In an in-circuit emulation system having a device under test coupled to and operating in lock-step with a virtual processor, a~~ A method of handling conditional jumps in ~~the~~ a virtual processor operating in lock-step with a device under test, comprising:

detecting an I/O read data ~~instruction followed immediately by a conditional jump instruction~~ sent by said device under test, wherein said I/O read data is followed immediately by a conditional jump instruction;

after receipt of said I/O read data, computing a speculative conditional jump address before a condition for said conditional jump instruction is satisfied, ~~prior to receipt of said conditional jump instruction from the virtual processor to remain in lockstep execution with said device under test~~; ~~and~~

~~determining~~ after receipt of the I/O read data from the device under test, determining whether ~~to proceed with instruction execution at a next consecutive address or at the conditional jump address~~ a conditional jump condition is met; and

executing instruction based on said determination, such that said virtual processor remains in lock-step execution with said device under test.

15. (currently amended) The ~~in-circuit emulation system~~ method according to claim 14, ~~further comprising~~ wherein said executing comprises execution of [[a]]

next consecutive instruction in the event said conditional jump condition is not met.

16. (currently amended) The ~~in-circuit emulation system~~ method according to claim 14, ~~further comprising~~ wherein said executing comprises execution of an instruction at the speculative conditional jump address in the event the conditional jump condition is met.

17. (currently amended) The ~~in-circuit emulation system~~ method according to claim 14, wherein the conditional jump address is computed while the I/O read data are sent from the device under test to the virtual processor.

18. (currently amended) The ~~in-circuit emulation system~~ method according to claim 14, wherein the device under test sets a zero flag if an I/O read test condition is met.

19. (currently amended) The ~~in-circuit emulation system~~ method according to claim 18, wherein the jump condition is met if the zero flag is set.

20. (currently amended) The ~~in-circuit emulation system~~ method according to claim 14, wherein the virtual processor is implemented in a Field Programmable Gate Array.